《计算机系统结构》往年考题参考答案

**仅供参考！**

一、简答题

1. 指令流水计算机中，采用独立的指令缓存与数据缓存对系统性能有什么好处。

【答】 ①解决访存和取指的结构冲突，加速流水线；②数据和指令的访问有所不同，设置单独的 Cache 有利于提高命中率，减少平均访存时间。

2. 什么是指令动态调度？使用寄存器重命名能够解决哪些数据冲突？

【答】指令动态调度是指在程序执行过程中，依靠专门的硬件队代码进行调度。使用寄存器重命名能够解决名相关（反相关、输出相关）。

3. 从数据和指令的角度，分别说明引起时间与空间局部性的原因。

【答】数据：对于刚被访问过的数据，很有可能再次访问它或者它附近的数据。指令：顺序执行的可能性较大。

4. 直接用虚拟地址索引缓存会存在什么问题？

【答】多个虚拟地址可能指向同一物理地址。

5. 多处理机为什么要维护缓存一致性？

【答】每个处理机都有一份自己的Cache，但是这些Cache都映射同一个物理主存，当一个处理机修改自己Cache的数据时，其他处理机中该部分的数据就不是最新的，造成不一致性。

二、填空题

1. 16个处理器组成的网络，使用洗牌函数相联，那么与10号相联的是 。

【答】10=(1010)2，循环左移一位得(0101)2=5。

2. 有16个处理器，编号为0,1,…,15，先经过PM2+3，再经过混洗变换后，11号处理器连向\_\_\_号处理器。

【答】11经过PM2+3得到，3再经过混洗变换得6。

3. 使用混洗交换单级网络将一个PE中的数据播送到所有16个PE中，需要\_\_\_\_次交换，需要\_\_\_混洗。假设每步只能进行混洗或交换中的一种变换。

【答】4次交换，3次混洗。

4. 16个处理器组成的网络，采用PM2±0，PM2±2链接，网络直径为 ，结点度为 。

【答】有结论：个处理器组成的网络，采用PM2±0，PM2±*n*/2链接，实现各处理单元之间上下左右互联。网络直径为3，结点度为4（？）。

5. 可以在向量与标量工作模式中切换的处理器，处理向量时效率是处理标量的9倍。已知运行一段程序时有1/4的时间在运行向量指令，向量指令的比例为 。

【答】设向量指令占比，则，解得。

6. 向量处理器在串行模式执行以下指令需要 拍，使用链接技术需要 拍。

v3 <- A (load, 6拍)

v2 <- v0 + v1 (add, 6拍)

v4 <- v2 \* v3 (mul, 7拍)

【答】串行模式下需要拍；

使用链接技术需要拍。

7. 处理器P1和P2执行A, B, C三种指令的周期如下

|  |  |  |
| --- | --- | --- |
|  | P1 | P2 |
| A | 1 | 2 |
| B | 2 | 3 |
| C | 4 | 4 |

一段程序中A占60%，B占30%，C占10%，分别求P1和P2运行该程序时的CPI。

【答】P1运行该程序的CPI为，P2运行该程序的CPI为。

8. 已知一处理器指令缓存不命中率为2%，数据缓存不命中率为4%，不命中代价为100周期。命中时，CPI为2，那么执行一段含有Load/Save指令各15%的程序时，其CPI为 。

【答】每条指令出现不命中的概率是，故其CPI为。

9. 五段流水线CPU，各段延迟时间分别为2.2ns, 2.5ns, 2.2ns, 2.3ns, 2.3ns。连续执行10条指令，需要的时间为\_\_\_\_，该CPU最高频率为\_\_\_\_MHz。

【答】各段不等长的流水线计算公式参见教材58页（时空图如如3.8），需要的时间为

ns，最大周期为2.5ns，即频率为400MHz。

10. 采用预留算法实现的非线性流水线优化调度，其启动循环为(1,3)，则该流水线周期*P*为 ，调度后的禁止集为 。

【答】周期，。

11. 有一指令系统，共有7条指令。有两种类型，一种为寄存器－寄存器型，一种为寄存器－存储器型。指令字长为8位或16位，不同类型指令字长不同。要求变址范围－127到128。则该指令系统最多可以编址 个通用寄存器，这时，最多可以编址 个变址寄存器。

【答】考虑寄存器－寄存器型指令字长为8位，有3条指令，除去2位操作码，剩下各3位编址8个通用寄存器。寄存器－存储器型指令字长为16位，有4条指令，除去8位立即数，3位寄存器，4位操作码，剩下1位用来编址2个变址寄存器。（所给答案可能不是最优的，但是掌握这种指令的各个部分的长度如何计算的方法即可）

12. 在100次内存访问中，一级cache缺失10次，二级cache缺失5次。则一级cache的全局命中率为 ，二级cache的全局命中率为 。

【答】90%；95%。

13. 分别在以下条件时计算块地址0110的索引(index)，缓存有8块，主存有16块：

a) 二路组相联 ；

b) 直接映射 。

【答】二路组相联共8/2=4组，索引为2位。直接映射索引为3位。答案分别是10、110。

14. 缓存共有4块，每块1 byte，采用LRU策略。访问字序列0, 1, 4, 1, 0, 4在下列情况下的命中率分别是：

a) 直接映射 ；

b) 二路组相联 。

【答】直接映射

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 4 | 1 | 0 | 4 |
| 0 | 缺失(0) |  | 缺失(4) |  | 缺失(0) | 缺失(4) |
| 1 |  | 缺失(1) |  | 命中 |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |

命中率为1/6。

二路组相联

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 4 | 1 | 0 | 4 |
| 0A | 缺失(0) |  |  |  | 命中 |  |
| 0B |  |  | 缺失(4) |  |  | 命中 |
| 1A |  | 缺失(1) |  | 命中 |  |  |
| 1B |  |  |  |  |  |  |

命中率为1/2。

三、判断题：以下对MIPS架构CPU的各改进方案，哪些修改了系统结构（Architecture），哪些只修改了实现（Implementation）？填写A或者I。

(1) 将32位指令改为64位指令

(2) 加入指令Cache

(3) 增加流水线的段数

(4) 减去某些定向（forwarding）相关逻辑的实现

(5) 取消气泡

(6) 增加16个额外的通用寄存器

(7) 增加对某指令集的支持

【解】(1) A

(2) I

(3) I

(4) I

(5) I

(6) A

(7) A

四、解答题。

1. 设计了一种优化方案。

·优化后的时钟周期比未优化的快15%；

·未优化的取/存指令占总数的30%；

·优化后的取/存指令比未优化的少1/3，其它无变化；

·未优化的所有指令均用1个时钟周期；优化的取/存指令用2个时钟周期，其它指令用1个时钟周期。

（1）求优化方案的平均CPI；

（2）通过计算加速比，判断哪个方案速度更快？

【解】(1) 不妨设优化前共10条指令，取/存指令有3条。优化后的取/存指令减少了1条，即还有2条取/存指令，其他7条不变，则总指令数变为9条。

优化方案的平均CPI为。

(2) 假设我们考虑这10条指令运行的总时间。以未优化的1个时钟周期作为单位1，优化前的运行时间为10，优化后的总时间为，说明优化后的更快。但是如果考虑平均每条指令的运行时间，那么优化前的更快（因为优化后平均每条指令的运行时间大于1）。

2. 在有32个处理机的并行机上运行一段程序，获得加速比26，已知该程序只有两种运行方式：在所有32个处理机上同时运行，或者只能由一个处理机执行。请问程序中只能由一个处理机执行的部分占多大比例？

【解】设程序中只能由一个处理机执行的部分占，根据加速比定义有



3. 某指令系统，有三地址指令4条，单地址指令255条，零地址指令16条。其指令字长12位，地址码3位。请问扩展编码是否可行？如果单地址指令是254条呢？

【解】（作业题）三地址指令共需要9位地址码，所以剩下3位编码操作码，用4个码点编码三地址指令，剩余4个用于扩展。单地址指令操作码有9位，可提供个码点，用其中255个编码单地址指令，剩下1个用于扩展。这时零地址指令只有8个码点可用，所以扩展编码不可行。

如果单地址指令是254条，那么还剩下2个用于扩展零地址指令，零地址指令自身的最低3位各可以编码8条指令，共计16条。因此可以实现扩展编码。

4. 指令字长16位，有双地址指令、单地址指令、零地址指令。地址都是6位。双地址指令15条。单地址与零地址条数相同。

(1) 单地址与零地址指令最多能有多少条？

(2) 给这三种指令分配操作码。

【解】(1) 双地址指令地址码占12位，故操作码有4位，共16个码点，剩下1个码点作扩展。单地址指令的操作码最长10位，零地址指令最长16位。对于单地址指令来说，最多有条，零地址指令也必须是63条。

(2) 双地址指令：0000~1110；

单地址指令：1111 000000~1111 111110；

零地址指令：1111 111111 000000~1111 111111 111110。

5. 全相联Cache采用写直达策略。初始Cache为空。分别对按写分配和不按写分配两种策略，计算以下操作执行后的命中率。

Write Mem[100]

Write Mem[100]

Read Mem[200]

Write Mem[200]

Write Mem[100]

【解】（PPT上的题）按写分配：结果分别是缺失、命中、缺失、命中、命中，命中率为60%。

不按写分配：结果分别是缺失、缺失、缺失、命中、缺失，命中率为20%。

6. Cache采用组相连映像及变换。主存1MB，Cache 32KB，块大小64B，Cache分为8组。

(1) 写出主存地址和缓存地址的格式(写出各域及位数)；

(2) 若Cache的访问周期为20ns，命中率0.95，要使加速比大于10，主存的访问周期应大于多少？

【解】(1) Cache共有32KB/64B=512块，分为8路，共有512/8=64组。

主存地址：标识8位，索引6位，块内地址6位。

缓存地址：？。

(2) 设主存访问周期为，则加速比ns。

7. Cache有4块，每块4字，采用直接映像法。初始时Cache为空。访问的字地址序列为：0,7,12,9,16,8,17,0,12,2。求cache命中率。

【解】访问过程如下：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 7 | 12 | 9 | 16 | 8 | 17 | 0 | 12 | 2 |
| 0 | Miss  0~3 |  |  |  | Miss  16~19 |  | Hit | Miss  0~3 |  | Hit |
| 1 |  | Miss  4~7 |  |  |  | Hit |  |  |  |  |
| 2 |  |  |  | Miss  8~11 |  |  |  |  |  |  |
| 3 |  |  | Miss  12~15 |  |  |  |  |  | Hit |  |

命中率为40%。

8. 一段程序有1000条指令，每条指令平均访问存储器1.5次，一级Cache访问需要1ns，二级Cache访问需要10ns，主存访问需要100ns。这段程序运行完后共访问二级Cache 90次，访问主存27次。

(1) 求一级Cache和二级Cache命中率；

(2) 求存储器等效访问时间；

(3) 求每条指令因为访问存储器造成的平均延迟。

【解】(1) 程序一共访存次，一级Cache缺失了90次，故

一级Cache（全局和局部）命中率为；

二级Cache的局部命中率为；

二级Cache的全局命中率为。

(2) 平均访存时间ns。

(3) 每条指令因为访问存储器造成的平均延迟为ns。

9. 某系统Cache为4路组相联，Cache大小为16K字节，块大小为64字节。按写分配。对于如下代码：

int M[4096], i, j;

for (i = 0; i < 10; i++) {

for (j = 0; j < 4096; j++) {

M[j] = i + j;

}

}

(1) 当i=0时，发生的Cache缺失是属于什么类型的缺失？发生了多少次？（4分）

(2) 运行完这段代码，求整体缺失率。（4分）

【解】(1) 4096长度的整型数组为16K字节，每个块为64字节，对应16个整数。当i=0时，对M各元素均是首次访问，属于必然缺失。发生次。

(2) 当i>0时，不再发生缺失。总访存次数为40960次，故整体缺失率为。

10. 一个缓存，采用m路组相联，顺序访问一个元素大小和缓存块大小相等的数组，求数组长度N

a) >m

b) <m

且缓存采用

a) LRU

b) OPT

时的命中率。

【解】都是0，因为一个元素大小和缓存块大小相等，而每次访问一个元素时也只会载入一个块大小的数据到Cache，故每次访问都必然会缺失。

11. 有以下指令（假设第一个操作数为写回的寄存器）

N1: load r0 a

N2: add r1 r0

N3: load r2 b

N4: mul r3 r4

N5: and r4 r5

N6: add r2 r5

(1) 请列出所有可能的数据冲突与结构冲突。

(2) 假设该处理器一个周期仅能进行一次访存操作，画出其执行上述指令的时空图。

【解】(1) 数据冲突：

N1与N2: r0 RAW

N4与N5: r4 WAR

N3与N6: r2 WAW

结构冲突：读取内存与取指冲突。

(2) 假设数据冲突需要插一个气泡。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 1 | IF | ID | EXE | **MEM** | WB |  |  |  |  |  |  |  |  |
| 2 |  | IF | ID | stall | EXE | MEM | WB |  |  |  |  |  |  |
| 3 |  |  | IF |  | ID | EXE | **MEM** | WB |  |  |  |  |  |
| 4 |  |  |  |  |  | IF | ID | EXE | MEM | WB |  |  |  |
| 5 |  |  |  |  |  |  | **stall** | IF | ID | EXE | MEM | WB |  |
| 6 |  |  |  |  |  |  |  |  | IF | ID | EXE | MEM | WB |

**说明：本答案中几乎所有给出的时空图与教材和课件上的形式不一样，建议考试中严格遵循教材和课件的标准（如教材图3.8）！**

12. 某CPU指令的运行分为取指、译码、执行、写结果四个阶段，每段延迟均为5ns。

运行程序如下：

K1 MOV R1,#4; R1 <- 向量长度4

K2 Loop: MOV R2,A(R1); R2 <- A向量的一个元素

K3 ADD　R0,R2; R0 <- (R0)+(R2)

K4 DNE R1,Loop; R1 <- (R1)-1, 若(R1)!=0, 则转向Loop

K5 MOV SUM,R0; SUM <- (R0) ,保存结果

(1) 列出所有的数据相关。

(2) 采用预测转移不成功的静态分支预测法，画出流水线的时空图，求吞吐率、加速比、译码段的效率。

(3) 采用预测转移成功的静态分支预测法，画出流水线的时空图，求吞吐率、加速比、执行段的效率。

【解】(1) K1与K2: R1 RAW

K2与K3: R2 RAW

K1与K4: R1 WAW

K2与K4: R1 WAR

K3与K5: R0 RAW

(2) 这里都假设只有分支预测出错才会造成流水线暂停。

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | … | 16 | 17 | 18 | 19 | 20 |
| K1 | IF | ID | EXE | WB |  |  |  |  |  |  |  |  |
| K2 |  | IF | ID | EXE | WB |  |  |  |  |  |  |  |
| K3 |  |  | IF | ID | EXE | WB |  |  |  |  |  |  |
| K4 |  |  |  | IF | ID | EXE |  |  |  |  |  |  |
| K2 |  |  |  |  | stall | IF |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |  |  |  |  |  |
| K4 |  |  |  |  |  |  |  | IF | ID | EXE | WB |  |
| K5 |  |  |  |  |  |  |  |  | IF | ID | EXE | WB |

吞吐率ns-1，加速比，译码段的效率。

(3)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | … | 13 | 14 | 15 | 16 | 17 | 18 |
| K1 | IF | ID | EXE | WB |  |  |  |  |  |  |  |  |  |
| K2 |  | IF | ID | EXE | WB |  |  |  |  |  |  |  |  |
| K3 |  |  | IF | ID | EXE | WB |  |  |  |  |  |  |  |
| K4 |  |  |  | IF | ID | EXE |  |  |  |  |  |  |  |
| K2 |  |  |  |  | IF | ID |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |  |  |  |  |  |  |
| K4 |  |  |  |  |  |  |  | IF | ID | EXE | WB |  |  |
| K5 |  |  |  |  |  |  |  |  | stall | IF | ID | EXE | WB |

吞吐率ns-1，加速比，执行段的效率。

13. 计算，加法需要2个时钟周期，乘法需要4个时钟周期。

(1) 串行处理器，有1个加法单元，1个乘法单元，但不能同时工作，求总的时钟周期；

(2) SIMD处理器，有8个PE，标号为0~7，连接为单向环，初始时和所在的处理机标号为，每个PE向相邻的PE转移（数据传输）需要1个周期，问最小要多少个周期完成计算。

(3) 一个SISD流水线，S4的输出可以直接到输入。一个乘法指令顺序执行S1 S2 S3 S4

一个加法指令执行S1 S4。每个1个周期。

S1

S2

S3

S4

(a)求最短运行时间?

(b)画出流水线的时空图;

(c)求S4的利用率。

【解】(1)；

(2) 似乎今年没有讲到这里？

(3) 这里假设流水线不能同时做加法和乘法，并且二者进行切换时必须排空流水线。如果该流水线可以这样的话，结果应该更优一点。

(a) ；

(b)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | … | 13 | 14 | 15 | 16 | … | 22 | 23 |
| 1x | S1 | S2 | S3 | S4 |  |  |  |  |  |  |  |  |  |  |
| 2x |  | S1 | S2 | S3 | S4 |  |  |  |  |  |  |  |  |  |
| 3x |  |  | S1 | S2 | S3 | S4 |  |  |  |  |  |  |  |  |
| 4x |  |  |  | S1 | S2 | S3 |  |  |  |  |  |  |  |  |
| 5x |  |  |  |  | S1 | S2 |  |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10x |  |  |  |  |  |  |  | S4 |  |  |  |  |  |  |
| 1+ |  |  |  |  |  |  |  |  | S1 | S4 |  |  |  |  |
| 2+ |  |  |  |  |  |  |  |  |  | S1 | S4 |  |  |  |
| … |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9+ |  |  |  |  |  |  |  |  |  |  |  |  | S1 | S4 |

(c) 10次乘法和9次加法都用了S4，故利用率为19/23。

14. 预约表如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| S1 | √ |  |  |  | √ |
| S2 |  |  |  | √ | √ |
| S3 |  | √ | √ | √ |  |

(1) 求禁止集；

(2) 求初始冲突向量；

(3) 用预留算法实现优化调度，若流水线时钟周期t为30ns，求该流水线的最大吞吐率。

【解】(1) 禁止集为；

(2) 由禁止集得到初始冲突向量1011；

(3) 最小平均启动距离为3，最小启动循环为(3)，插入非计算延迟得到最优调度的预约表为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| S1 | √ |  |  |  | √ |
| S2 |  |  |  | √ | √ |
| S3 |  | √ | √ | √ |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | … |
| S1 | 1 |  |  | 2 | 1 |  | 3 | 2 |  |  | 3 | … |
| S2 |  |  |  | 1 | 1 |  | 2 | 2 |  | 3 | 3 | … |
| S3 |  | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 |  | … |
| S4 |  |  |  | 1 |  |  |  |  |  |  |  | … |
| D |  |  |  |  | 1 |  |  |  |  |  |  |  |

执行条指令的吞吐率为，最大为 ns。

15. 一条有4个功能段的非线性流水线，每个功能段的延迟时间都相等，它的预约表如下

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| S1 | √ |  |  |  |  |  | √ |
| S2 |  | √ |  |  |  | √ |  |
| S3 |  |  | √ |  | √ |  |  |
| S4 |  |  |  | √ |  |  |  |

(1) 求禁止集合和初始冲突向量；

(2) 画出状态图；

(3) 找出最小启动循环，求最小平均启动时间；

(4) 如果用上一问的启动循环连续完成10条指令，求实际的吞吐率；

(5) 用插入非计算延迟的方法可以得到最优调度，求最优调度的最大吞吐率。

【解】（前几问是PPT上的例子）

(1) 禁止集合，初始冲突向量101010。

(2) 第一轮：

初始冲突向量右移1位：010101∨101010=111111；

初始冲突向量右移3位：000101∨101010=101111；

初始冲突向量右移5位：000001∨101010=101011；

初始冲突向量右移大于等于7位：000000∨101010=101010。

第二轮：

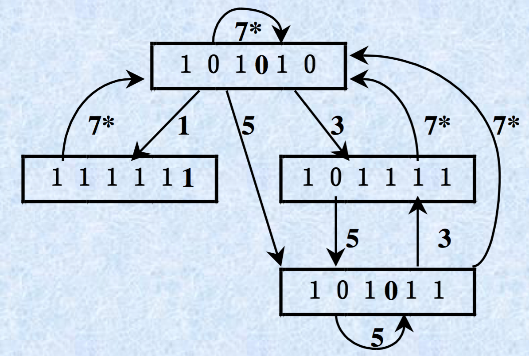
对于中间冲突向量111111，不做任何处理；

中间冲突向量101111右移5位：000001∨101010=101011；

中间冲突向量101011右移3位：000101∨101010=101111；

中间冲突向量101011右移5位：000001∨101010=101011。

画出状态图如下：



(3) 最小的启动循环为(1,7)和(3,5)，平均启动距离为 4。

(4) 采用循环(1,7)时，

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | … | 39 | 40 |
| S1 | 1 | 2 |  |  |  |  | 1 | 2 | 3 | 4 |  |  | … | 9 | 10 |
| S2 |  | 1 | 2 |  |  | 1 | 2 |  |  | 3 | 4 |  | … | 10 |  |
| S3 |  |  | 1 | 2 | 1 | 2 |  |  |  |  | 3 | 4 | … |  |  |
| S4 |  |  |  | 1 | 2 |  |  |  |  |  |  | 3 | … |  |  |

10条指令执行的总用时为40个周期，吞吐率为；

采用循环(3,5)时，

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | … | 18 | … | 42 |
| S1 | 1 |  |  | 2 |  |  | 1 |  | 3 | 2 |  | 4 | … | 4 | … | 10 |
| S2 |  | 1 |  |  | 2 | 1 |  |  | 2 | 3 |  |  | … | 5 | … |  |
| S3 |  |  | 1 |  | 1 | 2 |  | 2 |  |  | 3 |  | … |  | … |  |
| S4 |  |  |  | 1 |  |  | 2 |  |  |  |  | 3 | … |  | … |  |

10条指令执行的总用时为42个周期，吞吐率为。

(5) 最小平均启动距离为2，最小启动循环为(2)，插入非计算延迟得到最优调度的预约表为

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| S1 | √ |  |  |  |  |  |  | √ |
| S2 |  | √ |  |  |  |  | √ |  |
| S3 |  |  | √ |  |  | √ |  |  |
| S4 |  |  |  | √ |  |  |  |  |
| D |  |  |  |  | √ |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | … |
| S1 | 1 |  | 2 |  | 3 |  | 4 | 1 | 5 | 2 | 6 | 3 | 7 | 4 | … |
| S2 |  | 1 |  | 2 |  | 3 | 1 | 4 | 2 | 5 | 3 | 6 | 4 | 7 | … |
| S3 |  |  | 1 |  | 2 | 1 | 3 | 2 | 4 | 3 | 5 | 4 | 6 | 5 | … |
| S4 |  |  |  | 1 |  | 2 |  | 3 |  | 4 |  | 5 |  | 6 | … |
| D |  |  |  |  | 1 |  | 2 |  | 3 |  | 4 |  | 5 |  |  |

执行条指令的吞吐率为，最大为 。

16. 在一台每个时钟周期发射两条指令的超标量处理机上运行下面一段程序，所有指令都要经过"取指令"、"译码"、"执行"和"写结果"4个阶段，其中，"取指令"、"译码"和"写结果"三个阶段各为一个流水段，其延迟时间都为2ns。在"执行"阶段，LOAD操作和AND操作各延迟2ns，ADD操作延迟4ns，MUL操作延迟6ns，4种操作部件各设置一个。ADD部件和MUL部件都采用流水线结构，每一级流水线的延迟时间都为2ns。

n1: LOAD R0, A ; R0←主存(A)单元

n2: ADD R1, R0 ; R1←(R1)+(R0)

n3: LOAD R2, B ; R2←主存(B)单元

n4: MUL R3, R4 ; R3←(R3)×(R4)

n5: AND R4, R5 ; R4←(R4)&(R5)

n6: ADD R2, R5 ; R2←(R2)+(R5)

(1) 列出这个程序中所有的数据相关，包括写读数据相关、读写数据相关和写写数据相关。

(2) 如果所有运算型指令都在"译码"流水段读寄存器，在"写结果"流水段写寄存器，采用顺序发射顺序完成调度方法，画出流水线的时空图，并计算执行这个程序所用的时间。

(3) 如果所有运算型指令都在"译码"流水段读寄存器，在"写结果"流水段写寄存器，采用顺序发射乱序完成调度方法，画出流水线的时空图和各条指令完成的时间图，并计算执行这个程序所用的时间。

(4) 如果每个操作部件的输出端都有直接数据通路与输入端相连，采用顺序发射乱序完成调度方法，画出流水线的时空图和各条指令完成的时间图，并计算执行这个程序所用的时间。

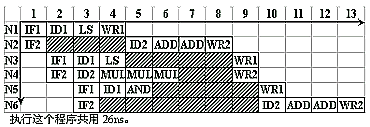
【解】(1)指令n1与n2之间有关于寄存器R0的写读数据相关，

指令n3与n6之间有关于寄存器R2的写读数据相关，

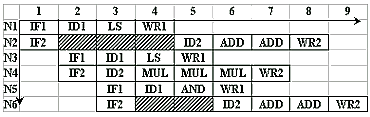
指令n4与n5之间有关于寄存器R4的读写数据相关，

指令n3与n6之间有关于寄存器R2的写写数据相关。

(2)采用顺序发射顺序完成调度方法的流水线时空图。



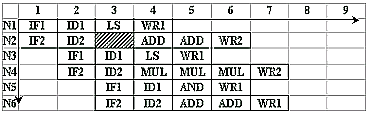
(3)采用顺序发射乱序完成调度方法的流水线时空图。



各条指令完成的时间图

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 时钟周期 | 4 | 5 | 6 | 7 | 8 | 9 |
| 流水线1 | n1 | n3 | n5 |  |  |  |
| 流水线2 |  |  |  | n4 | n2 | n6 |

|  |
| --- |
| 执行这个程序共用18ns。  (4)采用顺序发射乱序完成调度方法的流水线时空图。 |



各条指令完成的时间图

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 时钟周期 | 4 | 5 | 6 | 7 |
| 流水线1 | n1 | n3 | n5 | n6 |
| 流水线2 |  |  | n2 | n4 |

执行这个程序共用14ns。

17. 下面一段程序是计算浮点向量运算Y = a \* X + Y的，其中X和Y都是100维向量。采用循环展开的方式使得执行过程没有stall，那么最少需要展开几次？写出展开的程序。

LOOP: L.D F0,0(R1)

　　MUL.D F0,F0,F2

　　L.D F4,0(R2)

　　ADD.D F0,F0,F4

　　S.D F0,0(R2)

　　DSUBI R1,R1,#8

　　DSUBI R2,R2,#8

　　BNEZ R1,LOOP

注意：本题原来的题干有错，大家只要掌握循环展开的方法即可，一种可能的答案如下。本题在PPT上也有，可以看一下PPT上的题目和答案。

【解】将循环展开两次，进行指令调度，即可以消除延迟，代码如下：

　　LOOP： L.D F0，0（R1）

　　L.D F10，-8（R1）

　　MUL.D F0，F0，F2

　　MUL.D F10，F10，F2

　　L.D F4，0（R2）

　　L.D F14，-8（R2）

　　ADD.D F0，F0，F4

　　ADD.D F10，F10，F14

　　DSUBI R1，R1，#16

　　S.D 0（R2），F0

　　DSUBI R2，R2，#16

　　BNEZ R1，LOOP

S.D 8（R2），F10

假设：LOOP: L.D F0,0(R1)

(stall)

MUL.D F0,F0,F2

L.D F4,0(R2)

(stall)

ADD.D F0,F0,F4

(stall)

(stall)

　　S.D F0,0(R2)

　　DSUBI R1,R1,#8

　　DSUBI R2,R2,#8

BNEZ R1,LOOP

(stall)

【答2】另一个版本的答案，不知道哪个更优，就粘过来了。。

后执行的指令需要先执行的指令的结果需要经过的周期数如下：

先执行的 后执行的 周期数

FP ALU FP ALU 3

FP L.D FP ALU 2

FP ALU FP S.D 1

FP L.D FP S.D 0

BRANCH ANY 1

INT ALU INT ALU 0

INT ALU BRANCH 1

？？？？？？？？？？？？？？？？？？？？？

LOOP:

L.D F0,0(R1)

Stall

Stall

　　MUL.D F0,F0,F2

L.D F4,0(R2)

Stall

Stall

ADD.D F0,F0,F4

Stall

　　S.D F0,0(R2)

　　DSUBI R1,R1,#8

DSUBI R2,R2,#8

Stall

BNEZ R1,LOOP

Stall

两次：

L.D F0,0(R1)

Stall

Stall

　　MUL.D F0,F0,F2

L.D F4,0(R2)

Stall

Stall

ADD.D F0,F0,F4

Stall

S.D F0,0(R2)

L.D F6,-8(R1)

Stall

Stall

　　MUL.D F6,F6,F2

L.D F8,-8(R2)

Stall

Stall

ADD.D F6,F6,F8

Stall

S.D F6,-8(R2)

　　DSUBI R1,R1,#16

DSUBI R2,R2,#16

Stall

BNEZ R1,LOOP

Stall

优化调度：

L.D F0,0(R1)

L.D F4,0(R2)

　　MUL.D F0,F0,F2

　 MUL.D F6,F6,F2

L.D F6,-8(R1)

L.D F8,-8(R2)

ADD.D F0,F0,F4

ADD.D F6,F6,F8

DSUBI R1,R1,#16

S.D F0,0(R2)

DSUBI R2,R2,#16

BNEZ R1,LOOP

S.D F6,-8(R2)

18. 分支预测。

(1) 画出2位饱和计数器的状态图。

(2) 已知如下指令序列

|  |  |  |
| --- | --- | --- |
| 地址 | 目标地址 | 是否跳转 |
| \*\*\*\*01 | b1 | 否 |
| \*\*\*\*01 | b1 | 否 |
| \*\*\*\*01 | b1 | 是 |
| \*\*\*\*10 | b2 | 否 |
| \*\*\*\*10 | b2 | 是 |

已知初始BHT历史为00，BHT项全为01，求执行完上述程序后的BHT。

(3) 简要说明为何引入BTB会使得CPI下降。

【解】(1) 见教材134页图5.10。

(2) BHT一般就用PC的后几位作为表格的索引，查出里面的预测位。这里的预测当长度为2时就是上题中的饱和计数器，用00、01、10、11分别表示strongly not taken、weakly not taken、

weakly taken、strongly taken。可参考某中文blog<http://blog.csdn.net/edonlii/article/details/8754724>。执行完上述程序后的BHT为

|  |  |
| --- | --- |
| 索引 | 值（计数器） |
| 01 | 01 |
| 10 | 01 |

(3) 在经典的5段流水线中，ID段末尾才能获得分支目标地址。而采用BTB可以提前在IF段就知道这些信息，分支开销就可以减少为0。

19. 一个含有8个输入端的系统采用三层开关链接，使用开关控制。（可参见教材286页图9.21）。如开关处在0，则会不交换，如开关为1，则会发生交换。

(1) 若开关处在000状态，则0号链接？

(2) 若最左开关为0，那么1号不可能链接到哪些处理器？

【解】(1) 0号；

(2) 分类讨论后面两个开关的状态，得出1号的变换图：



1号不可能链接到4~7号处理器。

20. 在多处理机系统中，采用写回法的写作废 (write invalidate) 总线监听协议，参见教材310页图10.6的状态机。

(1) 给出M、I、S状态的定义，并说明什么时候可以确定发生了Cache不一致的情况。

(2) 假设有两个地址A和B（映射到不同的Cache块中），两个处理机P1和P2，初始时Cache全为空，根据特定的访问序列，补全下表（无消息用'-'代替，CPU事件RdM = Read Miss，RdH = Read Hit，WrM = Write Miss，WrH = Write Hit；总线消息WrMs = Write Miss，RdMs = Read Miss）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 操作 | A | | B | | 消息/操作 | |
| P1 | P2 | P1 | P2 | P1 | P2 |
| P1: R A | S | I | I | I | RdM/RdMs | RdMs/- |
| P2: W A 10 |  |  |  |  |  |  |
| P2: R A |  |  |  |  |  |  |
| P1: W A 20 |  |  |  |  |  |  |
| P1: W B 10 |  |  |  |  |  |  |
| P2: W B 20 |  |  |  |  |  |  |

【解】(1) M表示该块已经被修改过，并且未写入存储器。I表示该块内容无效。S表示该块要么处于共享状态，即多个处理器中都有其副本且都与存储器相同，或者该块只在一个处理器中有副本且与存储器相同。

(2)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 操作 | A | | B | | 消息/操作 | |
| P1 | P2 | P1 | P2 | P1 | P2 |
| P1: R A | S | I | I | I | RdM/RdMs | RdMs/- |
| P2: W A 10 | I | M | I | I | WrMs/- | WrM/WrMs |
| P2: R A | I | M | I | I | -/- | RdH/- |
| P1: W A 20 | M | I | I | I | WrM/WrMs | WrMs/写回 |
| P1: W B 10 | M | I | M | I | WrM/WrMs | WrMs/- |
| P2: W B 20 | M | I | I | M | WrMs/写回 | WrM/WrMs |

问题：

这些I M S到底是维护谁的变化？是A、B指向的cache block？还是P1、P2处理器的Cache？

A和B是互不影响？一个处理器不是只有一个cache吗？为什么在访问A和B的情况下，P1的cache不是一起变？ 为什么在访问A时，P1的cache状态变了，P2的没变？不都是A对应的cache吗？？

这些I M S到底是维护哪一个Cache block的变化？是说，P1和P2虽然都访问地址A，但是访问的Cache block不同吗？

如果A和B映射到相同的Cache？

PPT第42页